IN THE CLAIMS:

1. **(Currently Amended)** A PKI certificate architecture for a network connected gaming system, the gaming system including a plurality of gaming machines each having a plurality of executable software components, wherein each different executable software component within each gaming machine within the gaming system subject to receive certification is uniquely associated with a unique identifier and is signed with a separate and unique PKI certificate, the separate and unique PKI certificate being uniquely identified at least by the unique identifier, ~~**and**~~ wherein identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are signed with identical PKI certificates**, such that non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are signed with separate and different PKI certificates, and such that no two non-identical executable software components in different gaming machines are signed with a same PKI certificate**.

2. **(Previously Presented)** A PKI certificate architecture according to claim 1, wherein each software component is authorized by a regulatory authority.

3. **(Previously Presented)** A PKI certificate architecture according to claim 1, wherein the separate and unique PKI certificate is produced by the certification lab, by the gaming system supplier or by the trusted party designated by the regulatory authority.

4. **(Previously Presented)** A PKI certificate architecture according to claim 1, wherein each software component is code signed by a certification lab, by a gaming system supplier or by a trusted party designated by the regulatory authority.

5. **(Previously Presented)** A PKI certificate architecture according to claim 1, wherein the separate and unique identifier is a certificate field selected from a "Subject" field, an "issued to" field, a "subject name" field, a "CommonName" field, a "provider" field or a "publisher" field.

6. **(Previously Presented)** A PKI certificate architecture according to claim 1, wherein the unique identifier comprises at least one of fields and field extensions.

7. **(Previously Presented)** A PKI certificate architecture according to claim 1, wherein the unique identifier comprises at least one of a plurality of fields selected from among:

a software component part number;

a software component major version number;

a software component minor version number;

a software component build number;

a software component revision number;

a software component project name;

a software component type of software component;

a software component language variant;

a software component game regulation variant;

a software component friendly name;

an identification of the certification laboratory, and

an identification of the client.

8. **(Previously Presented)** A PKI certificate architecture according to claim 7, wherein the unique identifier is a concatenation of selected identifiers.

9. **(Previously Presented)** A PKI certificate architecture according to claim 1, wherein at least a portion of the unique identifier is reported in the Windows event log upon execution of the software component.

10. **(Previously Presented)** A PKI certificate architecture according to claim 1, wherein at least a portion of the unique identifier is reported in the source field of the Windows event log upon execution of the software component.

11. **(Previously Presented)** A PKI certificate architecture according to claim 1, wherein at least a portion of the unique identifier is reported in the Windows event log upon execution of the software component in a predetermined event log bin upon execution of the software component.

12. **(Previously Presented)** A PKI certificate architecture according to claim 1, wherein at least a portion of the unique identifier is traceable in at least one of:

source code;

Windows File Properties;

Trusted Inventory;

Windows Event Log;

Software Restriction Policies, and

Certificate Store.

13. **(Original)** A PKI certificate architecture according to claim 1, wherein the network connected gaming system is connected in at least one of a local area system and wide area network.

14. **(Previously Presented)** A PKI certificate architecture according to claim 1, wherein the network connected gaming system comprises at least one of gaming terminals, gaming servers and computers.

15. **(Previously Presented)** A PKI certificate architecture according to claim 1, wherein the unique identifier contains identification information delimited with file-name-allowed non-alphanumeric characters to facilitate human identification, string searches and file searches.

16. **(Previously Presented)** A PKI certificate architecture according to claim 1, wherein a selected set of identification information making up the unique identifier are used for making up the file name of PKI certificate related files such as *.CER, *.P7B and *.PVK such as to facilitate human identification, string searches and file searches.

17. **(Currently Amended)** A method for a network connected gaming system to prevent unauthorized software components of constituent computers of the gaming system from executing, the gaming system including a plurality of gaming machines each having a plurality of executable software components, the method comprising the steps of:

producing a separate and unique PKI certificate for each of the plurality of executable software component subject to receiving certification within each gaming machine, each software component subject to receiving certification including a unique identifier;

code signing each executable software component subject to receiving certification with its respective separate and unique PKI certificate, each respective PKI certificate being uniquely identified at least by a unique identifier that is uniquely associated with the executable software component such that identical executable software components in different ones of the plurality

of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates**, such that non-identical executable software components in different ones of the plurality of gaming machines are associated with separate and different identifiers and are code signed with separate and different PKI certificates and such that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate**, and

configuring software restriction policy certificate rules to allow execution of only those executable software components whose code signed PKI certificate is determined to be authorized.

18. **(Previously Presented)** A method according to claim 17, further comprising the step of configuring software restriction policy rules to prevent execution of unauthorized software components.

19. **(Previously Presented)** A method according to claim 17, further comprising the step of configuring software restriction policy rules to prevent execution of all not explicitly authorized software components.

20. (**Currently Amended)** A method for a network connected gaming system to enable only authorized software components of constituent computers of the gaming system to execute, comprising the steps of:

code signing each authorized software component with a PKI certificate such that identical authorized software components in different ones of the constituent computers are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the constituent computers are code signed with separate

and different PKI certificates and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate;

configuring a separate software restriction policy for each authorized software component in each of the constituent computers of the gaming system, and **associating the configured separate software restriction policy with the PKI certificate with which the authorized software component was code signed;**

enforcing the **associated** software restriction policy for each **code signed** authorized software component such that ~~the~~ each **code signed** authorized software component in each of the constituent computers of the gaming system must be authorized to run by its associated separate software restriction policy.

21. **(Previously Presented)** A method according to claim 20, wherein the authorized software components are mandated by a regulatory body.

22. **(Currently Amended)** A method for a network connected gaming system to enable only authorized software components of constituent computers of the gaming system to execute, comprising the steps of:

configuring a separate and unique certificate software restriction policy for each authorized executable software component of each of the constituent computers of the gaming system such that the each authorized executable software component in each of the constituent computers of the gaming system must be authorized to run by its associated separate software restriction policy;

code signing each authorized software component with a PKI certificate such that identical authorized software components in different ones of the constituent computers

are code signed with identical PKI certificates, such that non-identical authorized software components in different ones of the constituent computers are code signed with separate and different PKI certificates and such that no two non-identical authorized software components in different ones of the constituent gaming machines are code signed with a same PKI certificate;

configuring a path software restriction policy to prevent unauthorized software components from executing;

configuring a path software restriction policy to prevent non-explicitly authorized software components from executing;

enforcing the certificate software restriction policy configured for each **of the code signed** authorized executable software components of each of the constituent computers of the gaming system, and

enforcing the path software restriction policies.

23. **(Previously Presented)** A method according to claim 22, wherein the authorized software components are mandated by a regulatory body.

24. **(Currently Amended)** A method for a network connected gaming system to enable only authorized software components of constituent computers of the gaming system to execute, the gaming system including a plurality of gaming machines each having a plurality of executable software components, the method comprising the steps of:

producing a separate and unique PKI certificate for each of the plurality of executable software ~~**component**~~ **components** within the gaming system subject to receive certification, each respective PKI certificate being associated with a unique identifier that is uniquely associated with the **executable** software component such that identical executable software components in

different ones of the plurality of gaming machines of the network connected gaming system are associated with identical identifiers and are code signed with identical PKI certificates**, such that non-identical executable software components in different ones of the plurality of gaming machines are code signed with separate and different PKI certificates and such that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate**;

code signing each software component subject to receive certification with its respective separate and unique PKI certificate;

configuring a certificate software restriction policy for each of the respective separate and unique PKI certificates, and

enforcing the certificate software restriction policy for each of the respective separate and unique PKI certificates.

25. **(Currently Amended)** A method for downloading authorized executable software components and allowing execution of downloaded authorized executable software components of a plurality of gaming machines of a network connected gaming system, comprising the steps of:

for each of the plurality of gaming machines of the network connected gaming system:

code signing each authorized executable software component with a separate PKI certificate that is unique to the authorized software component such that identical executable software components in different ones of the plurality of gaming machines of the network connected gaming system are code signed with identical PKI certificates**, such that non-identical authorized software components in different ones of the plurality of gaming machines are code signed with separate and different PKI certificates and such that no two**

non-identical authorized software components in different gaming machines are code signed with a same PKI certificate;

packaging the code signed authorized software components into an installation package;

configuring install policies to install each code signed authorized executable software component contained in the installation package;

configuring certificate rule policies to allow execution of the installed code signed authorized executable software component;

configuring enforcement of the policies.

26. **(Withdrawn)** A method for a network connected gaming system to enable selective execution of at least one authorized software component, comprising the steps of:

configuring Software Restriction Policies for the at least one authorized software component at a predetermined time;

unrestricting the Software Restriction Policies for the at least one authorized software component at a predetermined time;

enabling a link for the Software Restriction Policies for the at least one authorized software component at a predetermined time;

checking for a change of the Software Restriction Policies and if there is no policy change then looping to the beginning of this step, and

enforcing the change of the Software Restriction Policies at a predetermined time.

27. **(Withdrawn)** A method for a network connected gaming system according to claim 26, wherein the checking step includes checking for the change of the Software Restriction

Policies whenever a predetermined timeout has expired subsequent to the player balance reaching zero and if there is no policy change then looping to the beginning of this step.

28. **(Withdrawn)** A method for a network connected gaming system according to claim 26, further comprising the step of displaying a list of authorized software to the player for selection.

29. **(Withdrawn)** A method for a network connected gaming system according to claim 26, wherein a rule for the Software Restriction Policies is at least one of certificate rule, path rule, hash rule, Internet zone rule and registry path rule.

30. **(Withdrawn)** A method for a network connected gaming system according to claim 26, wherein the network connected gaming system is connected in at least one of a local area system and a wide area network.

31. **(Withdrawn)** A method for a network connected gaming system according to claim 26, wherein the network connected gaming system comprises at least one of gaming terminals and gaming servers.

32. **(Withdrawn)** A method for a network connected gaming system according to claim 26, wherein the checking step includes executing the RegisterGPNotification function.

33. **(Withdrawn)** A method for a network connected gaming system according to claim 26, wherein the checking step is bypassed.

34. **(Withdrawn)** A method for a network connected gaming system according to claim 26, wherein the enforcing step includes executing the gpupdate function.

35. **(Withdrawn)** A method for a network connected gaming system according to claim 26, wherein the enforcing step includes executing the gpupdate function followed by a reboot.

36. **(Withdrawn)** A method for a network connected gaming system according to claim 26, wherein the enforcing step includes executing the RefreshPolicy or RefreshPolicyEx function.

37. **(Withdrawn)** A method for a network connected gaming system according to claim 26, wherein the enforcing step includes executing the RefreshPolicy or RefreshPolicyEx function followed by a reboot.

38. **(Withdrawn)** A method for a network connected gaming system according to claim 26, further comprising the steps of:

configuring Software Installation Policies for the at least one authorized software component at a predetermined time;

enabling a link for the software installation policies for the at least one authorized software component at a predetermined time;

checking for a change of the Software Installation Policies and if there is no policy change then looping to the beginning of this step, and

enforcing the change of the software installation policies.

39. **(Withdrawn)** A method for a network connected gaming system according to claim 38, wherein the checking step includes checking for the change of the software installation policies whenever a predetermined timeout has expired subsequent to the player balance reaching zero and if there is no policy change then looping to the beginning of this step.

40. **(Withdrawn)** A method for a network connected gaming system according to claim 38, wherein the checking step includes executing the RegisterGPNotification function.

41. **(Withdrawn)** A method for a network connected gaming system according to claim 38, wherein the checking step is bypassed.

42. **(Withdrawn)** A method for a network connected gaming system according to claim 38, wherein the enforcing step includes executing the gpupdate function.

43. **(Withdrawn)** A method for a network connected gaming system according to claim 38, wherein the enforcing step includes executing the gpupdate function followed by a reboot.

44. **(Withdrawn)** A method for a network connected gaming system according to claim 38, wherein the enforcing step includes executing the RefreshPolicy or RefreshPolicyEx function.

45. **(Withdrawn)** A method for a network connected gaming system according to claim 38, wherein the enforcing step includes executing the RefreshPolicy or RefreshPolicyEx function followed by a reboot.

46. **(Withdrawn)** A method for a network connected gaming system according to claim 38, further comprising the step of displaying a list of authorized software to the player for selection.

47. **(Withdrawn)** A method for a network connected gaming system according to claim 26, further comprising the initial steps of:

monitoring the game activity of players, and

choosing the at least one authorized software components in order to adapt game offering on the gaming terminals.

48. **(Withdrawn)** A method for a network connected gaming system according to claim 47, wherein the monitoring and choosing steps are carried out in a close-loop fashion such as to optimize player game activity in real-time.

49. **(Withdrawn)** A method for a network connected gaming system to enable selective availability of games on gaming terminals, comprising the steps of:

installing a plurality of game software on a selected set of gaming terminals;

choosing a selected set of installed game software to offer to players of the gaming terminals;

a first activating the chosen selected set of installed game software on a selected set of gaming terminals;

monitoring the game activity of the players on a selected set of gaming terminals;

modifying the selected set of installed game software to offer to players;

a second activating the modified selected set of installed game software on a selected set of gaming terminals;

50. **(Withdrawn)** A method for a network connected gaming system according to claim 49, wherein the monitoring, modifying and activating steps are executed in a close-loop fashion such as to optimize player game activity in real-time.

51. **(Withdrawn)** A method for a network connected gaming system according to claim 49, further comprising the step of displaying a list of authorized software to the player for selection.

52. **(Withdrawn)** A method for a network connected gaming system according to claim 49, further comprising a step of downloading at least one authorized game software to a selected set of the of gaming terminals;

53. **(Withdrawn)** A method for a network connected gaming system to enable selective availability of games on PC based gaming terminals, comprising the steps of:

selecting game software to be made available to players on a selected set of gaming terminals;

terminating all gaming software on a selected set of gaming terminals to transform each gaming terminals into a generic PC communicating in the network connected gaming system;

downloading via the network the selected game software to the generic PCs, and

starting the game software to transform the generic PCs into gaming terminals.

54. **(Withdrawn)** A method for a network connected gaming system according to claim 53, further comprising the step of displaying an "out-of-service" message or equivalent message to the player while the gaming terminal is transformed into a generic PC.

55. **(Withdrawn)** A method for a network connected gaming system according to claim 53, further comprising the step of displaying a list of software to the player for selection.

56. **(Withdrawn)** A method for a network connected gaming system according to claim 53, wherein the game software is authorized by a regulatory authority.

57. **(Withdrawn)** A method for a network connected gaming system according to claim 53, wherein booting is at least one of cold-booting, hot-booting and power-on booting.

58. **(Withdrawn)** A method for a network connected gaming system according to claim 53, wherein the PC based gaming terminals run a version of the Microsoft Windows operating system

59. **(Withdrawn)** A method for a network connected gaming system according to claim 53, wherein the step of downloading game software uses the Software Installation Policy (SIP) feature of the Windows operating system.

60. **(Withdrawn)** A method for a network connected gaming system according to claim 53, wherein the step of downloading game software uses the Microsoft SMS Systems Management Server.

61. **(Withdrawn)** A method for a network connected gaming system according to claim 53, further comprising the step of preventing unauthorized software from executing using the Software Restriction Policy feature.

62. **(Withdrawn)** A method for a network connected gaming system to enable selective availability of games on PC based gaming terminals, comprising the steps of:

selecting game software to be made available to players on a selected set of gaming terminals;

terminating all gaming software on a selected set of gaming terminals to transform each gaming terminal into a generic PC communicating in the network connected gaming system;

booting the generic PCs;

starting an operating system on the generic PCs;

downloading via the network the selected game software to the generic PCs, and

starting the game software to transform the generic PCs into gaming terminals.

63. **(Withdrawn)** A method for a network connected gaming system according to claim 62, further comprising the step of displaying an "out-of-service" message or equivalent message to the player while the gaming terminal is transformed into a generic PC.

64. **(Withdrawn)** A method for a network connected gaming system according to claim 62, further comprising the step of displaying a list of software to the player for selection.

65. **(Withdrawn)** A method for a network connected gaming system according to claim 62, wherein the game software is authorized by a regulatory authority.

66. **(Withdrawn)** A method for a network connected gaming system according to claim 62, wherein booting is at least one of cold-booting, hot-booting and power-on booting.

67. **(Withdrawn)** A method for a network connected gaming system according to claim 62, wherein PC based gaming terminals run a version of the Microsoft Windows operating system.

68. **(Withdrawn)** A method for a network connected gaming system according to claim 62, wherein the step of downloading game software uses the Software Installation Policy feature of the Windows operating system.

69. **(Withdrawn)** A method for a network connected gaming system according to claim 62, further comprising the step of preventing unauthorized software from executing using the Software Restriction Policy feature.

70. **(Withdrawn)** A method for a network connected gaming system according to claim 62, wherein the step of downloading game software uses the Microsoft SMS Systems Management Server.

71. **(Previously Presented)** A method for a network connected gaming system to prevent unauthorized executable files of constituent computers of the gaming system from executing, comprising the steps of:

packaging the authorized executable files into a code signed installation package;

configuring certificate rule policies to enable execution of the code signed installation package;

enforcing the policies, and

executing the code signed installation package upon every startup of any of the constituent computers of the gaming system or upon a command, wherein execution of any authorized executable file is predicated upon successfully executing the code signed installation package into which the authorized executable file is packaged.

72. **(Previously Presented)** A method according to claim 71, wherein the code signing uses a separate and unique PKI certificate for each installation package.

73. **(Previously Presented)** A method for a network connected gaming system to prevent unauthorized executable code of constituent computers of the gaming system from executing, comprising the steps of:

packaging the authorized executable files into a code signed installation package;

configuring certificate rule policies to enable execution of the code signed installation package;

configuring enforcement of the policies, and

re-installing the code signed installation package at every startup of any of the constituent computers of the gaming system or upon a command, wherein execution of any authorized

executable file is predicated upon successfully executing the code signed installation package into which the authorized executable file is packaged.

74. **(Previously Presented)** A method according to claim 73, wherein the code signing uses a separate and unique PKI certificate for each installation package.

75. **(Previously Presented)** A method for a network connected gaming system to prevent data of unauthorized non-executable files of constituent computers of the gaming system from affecting game outcome, comprising the steps of:

packaging the non-executable files into a code signed installation;

configuring certificate rule policies to enable execution of the code signed installation package;

configuring enforcement of the policies, and

executing the code signed installation package upon every startup of any of the constituent computers of the gaming system or upon a command.

76. **(Previously Presented)** A method according to claim 75, wherein the code signing uses a separate and unique PKI certificate for each installation package.

77. **(Previously Presented)** A method for trusting at least one authorized non-executable software component certified to comply with regulatory requirements downloaded into a network connected gaming system, the gaming system including a plurality of computers, the method comprising the steps of:

packaging the at least one non-executable file into at least one code signed installation package;

configuring certificate rule policies to enable execution of the at least one code signed installation package;

configuring enforcement of the policies, and

re-installing the at least one code signed installation package at every startup of any of the constituent computers of the gaming system or upon a command.

78. **(Previously Presented)** A method according to claim 77, wherein the at least one code signed package includes a separate and unique PKI certificate for each of the at least one installation package.

79. **(Previously Presented)** A method for scheduling at least one authorized executable software component installed in a network connected gaming system, the gaming system including a plurality of gaming machines, the method comprising the steps of:

packaging at least one authorized non-executable file that controls the scheduling of the at least one authorized executable software component into at least one code signed installation package, each of the at least one code signed installation packages including a predetermined PKI certificate;

configuring certificate rule policies to enable execution of the at least one code signed installation package in selected ones of the plurality of gaming machines; and

configuring enforcement of the certificate rule policies; and

downloading the at least one code signed installation package into the selected ones of the plurality of gaming machines;

executing the at least one code signed installation package.

80. **(Canceled)**

81. **(Previously Presented)** A method for scheduling at least one authorized executable software component according to claim 79, further comprising the step of re-installing the at least one code signed installation package at every startup of any of the constituent gaming machines of the gaming system or upon a command.

82. **(Currently Amended)** An automated platform to enable an on-going regulatory certification of a plurality of authorized software components of a network connected gaming system including a plurality of computers, the method comprising:

a reference platform representative of a target network connected gaming system and comprising a software-building environment located at a manufacturer or subcontractor of the software components;

a certification platform located at a regulatory certification authority, the certification platform being substantially identical to the reference platform, and

code-signing means for enabling the manufacturer or subcontractor to associate a separate and unique PKI certificate with each authorized software component subject to regulatory certification such **that** identical authorized software components subject to regulatory certification in different ones of the plurality of gaming machines of the network connected gaming system are code signed with identical PKI certificates**, such that non-identical executable software components in different ones of the plurality of gaming machines are code signed with separate and different PKI certificates, and such that no two non-identical executable software components in different gaming machines are code signed with a same PKI certificate**.

83. **(Previously Presented)** An automated platform according to claim 82, further comprising a secure communication link between the reference platform and the certification

platform for enabling manufacturer or designated subcontractors to remotely configure the software building environment on the certification platform.

84. **(Previously Presented)** An automated platform according to claim 82, wherein the authorized software components to be downloaded to the network connected gaming system are tested by the certification laboratory.

85. **(Previously Presented)** An automated platform according to claim 82, wherein the authorized software components to be downloaded to the network connected gaming system are compiled by the certification laboratory.

86. **(Previously Presented)** An automated platform according to claim 82, further comprising a secure communication link between the reference platform and the certification_for enabling remote assistance.

87. **(Previously Presented)** An automated platform according to claim 82, further comprising a secure communication link between the reference platform and the certification_for enabling users to carry out certification steps from a remotely located computer.

88. **(Original)** An automated platform according to claim 82, wherein the code signing means comprises a certificate authority under control of the manufacturer for generating certificates.

89. **(Original)** An automated platform according to claim 82, wherein the code signing means comprises a certificate authority under control of the regulatory certification authority for generating certificates.

90. **(Previously Presented)** An automated platform according to claim 82, further comprising means for maintaining the software-building environment of the reference platform and the software-building environment of the certification platform synchronized.

91. **(Withdrawn)** A method for a gaming terminal in a network connected gaming system to generate a list of authorized games available to the players comprising the steps of:

enforcing Software Restriction Policy for preventing non-authorized software components from executing;

enforcing Software Restriction Policy for enabling execution of a selected set of authorized games;

attempting to execute each game, and

adding games that have not been denied execution to a menu list.

92. **(Withdrawn)** A method for a network connected gaming system–according to claim 91, further comprising the step of removing games from the menu list for games that have been denied execution.

93. **(Withdrawn)** A method for a network connected gaming system according to claim 91, further comprising the step of removing games from the menu list for games whose executable file are not found .

94. **(Previously Presented)** A method for a gaming machine in a network connected gaming system to generate a menu of authorized games available to players, the method comprising the steps of:

generating a unique code signed PKI certificate for a predetermined software module of each authorized game;

generating an executable companion file for each authorized game, wherein the executable companion file is configured to execute faster than the authorized game;

code signing both the predetermined software module and its executable companion file with the generated PKI certificate;

enforcing software restriction policy rules for preventing non-authorized software components from executing;

enforcing software restriction policy rules for enabling execution of selected ones of the authorized games;

attempting to execute each executable companion file, and

adding only those games to the menu of authorized games whose executable companion file has not been denied execution by the software restriction policy rules.

95. **(Previously Presented)** A method according to claim 94 further comprising the step of removing games from the menu of authorized games whose companion file is denied execution by the software restriction policy rules.

96. **(Previously Presented)** A method according to claim 94, further comprising the step of removing games from the menu of authorized games whose companion executable file is not found.

97. **(Previously Presented)** A method for scheduling at least one authorized executable software component according to claim 79, wherein the code signing uses a separate and unique PKI certificate for each of the at least one installation package.